

Verantwoording Leerlijn programmeren, Wikiwijs arrangement

Deze leerlijn is tot stand gekomen vanuit een versnellingsvraag van het PO Doorbraakproject Slimmer Leren met ICT 2015-2016. Dankzij de inzet van de besturen [OPONOA](#) (Gelderland) en onderwijsgroep [FIER](#) (Friesland), met hulp van [SLO](#) en met ondersteuning van [Kennisnet](#).

Bij de totstandkoming van deze leerlijn stonden de besturen aan het roer. Op deze pagina wordt een aantal van de belangrijkste keuzes beschreven die zijn gemaakt tijdens het inhoudelijke proces om te komen tot een gestructureerd aanbod. Met als doel: iedereen kan aan de slag met programmeren in de klas!

Leerlijn programmeren als arrangement, startpunt of eindpunt

Deze leerlijn is als Wikiwijs arrangement beschikbaar gesteld. Dat stelt iedereen die dat maar wil in de gelegenheid om een eigen invulling te geven aan de lijn. En waar dat wenselijk is, aan te vullen, uit te breiden etc. Iedere school kan hierin eigen keuzes maken.

De activiteiten dienen als voorbeeld en inspiratie. Het arrangement is een verzameling begrippen, doelen en activiteiten en biedt de mogelijkheid om uit te breiden, met andere activiteiten en met online tools en software. Dit aanbod groeit dagelijks.

Iedere school kan hiermee eigen keuzes maken, afhankelijk van aanwezige voorkennis over programmeren, infrastructuur, devices en ambities. Bij tools en activiteiten kun je kijken welke elementen uit de leerlijn in de tools aan bod komen, om zo tot zonder enige verplichting tot een eigen invulling te kunnen komen.

Computational thinking

SLO definieert computational thinking als volgt:

“Computational thinking is het procesmatig (her)formuleren van problemen op een zodanige manier dat het mogelijk wordt om met computertechnologie het probleem op te lossen. Het gaat daarbij om een verzameling van denkprocessen waarbij probleemformulering, gegevensorganisatie, -analyse en -representatie worden gebruikt voor het oplossen van problemen met behulp van ICT-technieken en -gereedschappen.”

In het zgn. *voorbeeldmatig leerplankader* van SLO is een verdere, *voorbeeldmatige* uitwerking te vinden van de begrippen en doelen voor computational thinking. Deze te vinden op de [website van SLO](#).

Voor de leerlijn is gekozen om niet deze begrippen te kopiëren, maar om “eigen” begrippen te gebruiken. Deze keuze komt voort uit de gedachte om de leerlijn goed praktisch toepasbaar te maken voor leerkrachten, met eenduidige en herkenbare begrippen.

Wel hebben we bij elk begrip een verbinding proberen te leggen met de SLO computational thinking doelen. Programmeren is hier een onderdeel van. Bij elk begrip dat we behandelen is een pagina opgenomen waarin we aangeven welke computational thinking doelen

verbonden zijn aan het begrip. Zo werken leerlingen ook aan die begrippen en doelen, maar hanteert de leerlijn een eigen set begrippen toegespitst op programmeren.

Begrippen

De leerlijn programmeren is opgebouwd aan de hand van tien begrippen. Deze begrippen zien we als de “grammatica” van het programmeren. Met kennis en ervaring over deze concepten kun je met programmeren aan de slag. Bij elk begrip staat een korte toelichting ter verduidelijking van wat het begrip inhoudt.

We hebben daarbij nagedacht over de volgorde van de begrippen. De begrippen nemen qua complexiteit toe, onder andere door toenemende abstractie en doordat de begrippen meer en meer met elkaar in verbinding komen te staan.

Doelen

Voor de tien begrippen zijn doelen geformuleerd. De onderverdeling die daarbij is gehanteerd: onderbouw, middenbouw en bovenbouw. Per domein zijn waar relevant door ons specifieke doelen geformuleerd. Deze doelen borduren voort op de voorgaande doelen per begrip en per “bouw”.

Na onderbouw is het logisch om de doelen voor respectievelijk midden- en bovenbouw te behandelen. In de volgorde van de begrippen is gekeken of verdieping nodig is in elke “bouw”. Voor sommige begrippen belichten we uitsluitend doelen voor bijvoorbeeld onderbouw en middenbouw. Of alleen middenbouw-bovenbouw. Dit komt voort uit het idee dat sommige begrippen aan de orde komen bij andere doelen of geen verdieping meer behoeven. *Decompositie* bijvoorbeeld is een begrip dat vaker aan de orde zal komen bij de behandeling van de andere begrippen.

Uiteraard staat het iedereen vrij om hier eigen keuzes in te maken, door bijvoorbeeld te kiezen om in de onderbouw de activiteiten rond programmeren nog niet aan te bieden. En deze door te schuiven naar de middenbouw. Of om ervoor te kiezen om met leerlingen uit de middenbouw die meer willen doen eerder te gaan werken aan bovenbouw doelen.

Activiteiten

Bij de verschillende doelen en leeftijdsgroepen (onderbouw, middenbouw en bovenbouw) staan voorbeeldactiviteiten waarmee je werkt aan de geformuleerde doelen. Deze voorbeeldactiviteiten zijn zogenaamde “unplugged” activiteiten, waarmee je kunt werken aan het leren programmeren/computational thinking zonder daarbij veel voorkennis, bepaalde hardware, software en infrastructuur bij nodig te hebben.

Hierdoor zijn ze door iedereen te gebruiken met zo min mogelijk drempels.

Met dit materiaal leg je de basis voor programmeren, ongeacht welke programmeertaal of –tool men hierna zou willen gebruiken. De ontwikkelingen gaan snel: wat nu nog beschikbaar is, zou morgen bijvoorbeeld achterhaald en niet meer beschikbaar kunnen zijn.

Veel van deze voorbeeldactiviteiten zijn afkomstig van het code.org platform, een Amerikaanse website met activiteiten die beschikbaar zijn gesteld onder een CC-BY-NC-SA 4.0 licentie en zijn vertaald naar het Nederlands. Bij sommige activiteiten zijn aanpassingen gedaan om ze aan te passen aan de Nederlandse context, zoals bijvoorbeeld liedjes die gebruikt worden in een activiteit.